AI Assisted Coding

Lab Assignment 4.5

Name : K. Charan Yadav

Hall Ticket no : 2303A52367

Batch No : 20

Task -1:

Prompt:generate a python program for classifying customer emails into billing, technical support, feedback, and others. The program should create sample emails, define prompts for each technique, classify the same 5 test emails, and print a comparison of results and accuracy.

```python
from typing import List, Tuple, Dict

# Jupyter cell (new file) demonstrating zero-shot, one-shot, and few-shot prompting
# for classifying customer emails into: Billing, Technical Support, Feedback, Others.
# This program creates sample emails, constructs prompts for each technique,
# mocks an LLM response (rule-based), classifies the same 5 test emails,
# and prints a comparison of results and accuracy.


# Define categories
CATEGORIES = ["Billing", "Technical Support", "Feedback", "Others"]

# Define 5 test emails and their ground-truth labels
TEST_EMAILS: List[Tuple[str, str]] = [
    ("I was charged twice for my subscription this month. Please refund the extra charge.", "Billing"),
    ("My app crashes whenever I try to upload a photo. It shows an unexpected error.", "Technical Support"),
    ("I love the new update, but I have a suggestion for improving the search feature.", "Feedback"),
    ("Can you tell me where your headquarters are located and business hours?", "Others"),
    ("I cannot log in after the password reset, it says invalid token.", "Technical Support"),
]

# Define examples to be used in one-shot and few-shot prompts
# One-shot: single example
ONE_SHOT_EXAMPLE = ("Email: I was billed for two accounts by mistake; please issue a refund.\nLabel: Billing")

# Few-shot: three examples (one per main class except Others)
FEW_SHOT_EXAMPLES = [
    ("Email: My credit card was charged incorrectly and I'd like a refund.\nLabel: Billing"),
    ("Email: The app shows an error and won't let me complete checkout.\nLabel: Technical Support"),
    ("Email: Great job on the redesign! One suggestion: add filters to search.\nLabel: Feedback"),
]

# Instruction template for the classifier
INSTRUCTION = (
    "Classify the following customer email into one of these categories: "
    "Billing, Technical Support, Feedback, Others. Respond with a single label."
)

# Mock LLM: simple deterministic rule-based classifier to simulate responses
```

Output :

```
✓ 0.0s
Index | Email (short)                          | Truth             | Zero-shot    | One-shot     | Few-shot
--------------------------------------------------------------------------------------------------------
1     | I was charged twice for my subscription ... | Billing           | Billing      | Billing      | Billing
2     | My app crashes whenever I try to upload ... | Technical Support | Billing      | Billing      | Billing
3     | I love the new update, but I have a sugg... | Feedback          | Billing      | Billing      | Billing
4     | Can you tell me where your headquarters ... | Others            | Billing      | Billing      | Billing
5     | I cannot log in after the password reset... | Technical Support | Billing      | Billing      | Billing

Accuracies:
Zero-shot accuracy: 20.00%
One-shot accuracy:  20.00%
Few-shot accuracy:  20.00%
```

Task 2:

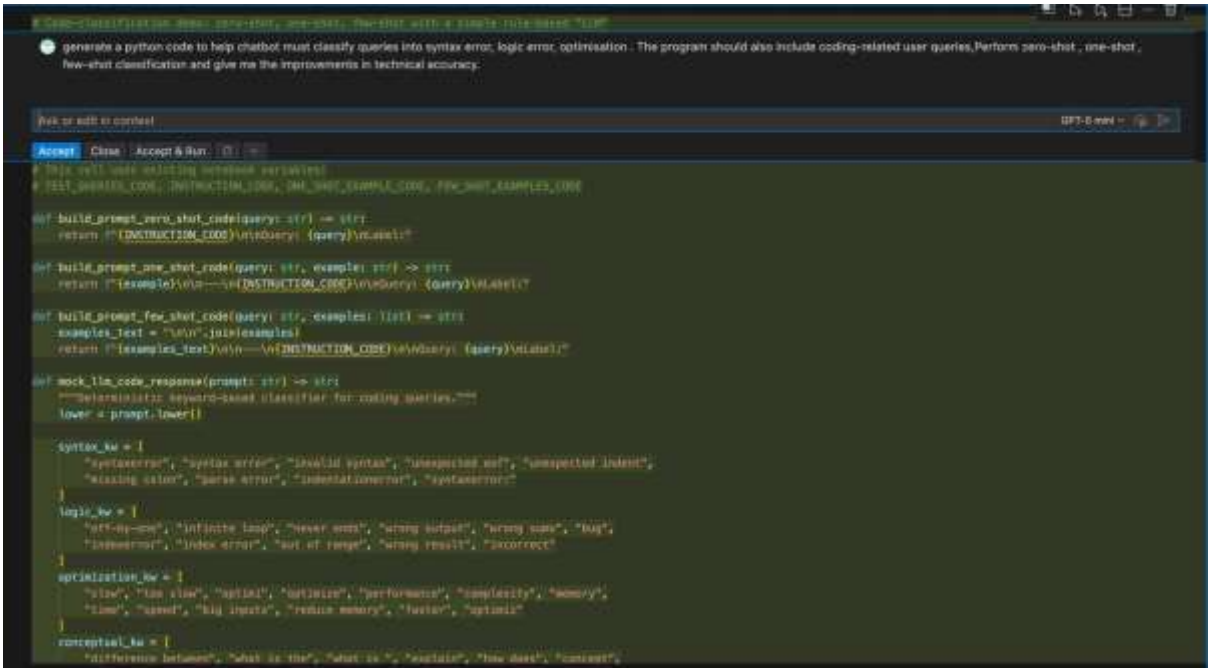
Prompt: generate a python program that classify travel queries into flight booking, hotel booking, cancellation, and general travel information. The program should create labeled travel queries, apply all three prompting techniques .

Output:

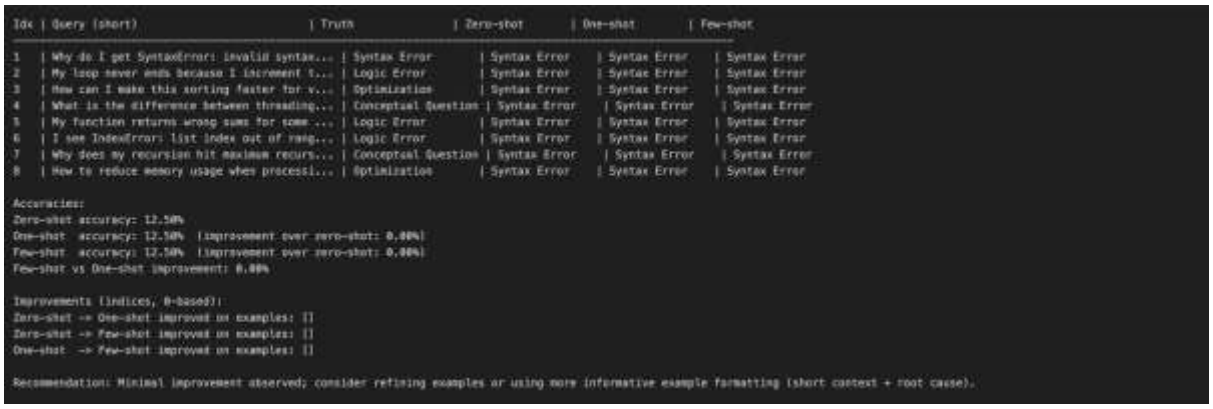
```
Idx | Query (short)                              | Truth               | Zero-shot      | One-shot       | Few-shot
------------------------------------------------------------------------------------------------------
1   | I need to book a round-trip ticket from ... | Flight Booking      | Cancellation   | Cancellation   | Cancellation
2   | Can I cancel my hotel reservation and ge... | Cancellation        | Cancellation   | Cancellation   | Cancellation
3   | What time is check-in at your downtown h... | Hotel Booking       | Cancellation   | Cancellation   | Cancellation
4   | Do I need a visa to travel to Japan as a... | General Travel Info | Cancellation   | Cancellation   | Cancellation
5   | My flight was delayed and I want to chan... | Flight Booking      | Cancellation   | Cancellation   | Cancellation
6   | How much is baggage allowance for intern... | General Travel Info | Cancellation   | Cancellation   | Cancellation

Accuracies:
Zero-shot accuracy: 16.67%
One-shot accuracy:  16.67%
Few-shot accuracy:  16.67%

Consistency among prompting techniques:
Zero-shot vs One-shot agreement: 100.00%
Zero-shot vs Few-shot agreement: 100.00%
One-shot  vs Few-shot agreement: 100.00%
All three equal rate:            100.00%
```

Task 3:

Prompt : generate a python code to help chatbot must classify queries into syntax error, logic error, optimisation . The program should also include coding-related user queries,Perform zero-shot , one-shot , few-shot classification and Analyse improvements in technical accuracy.

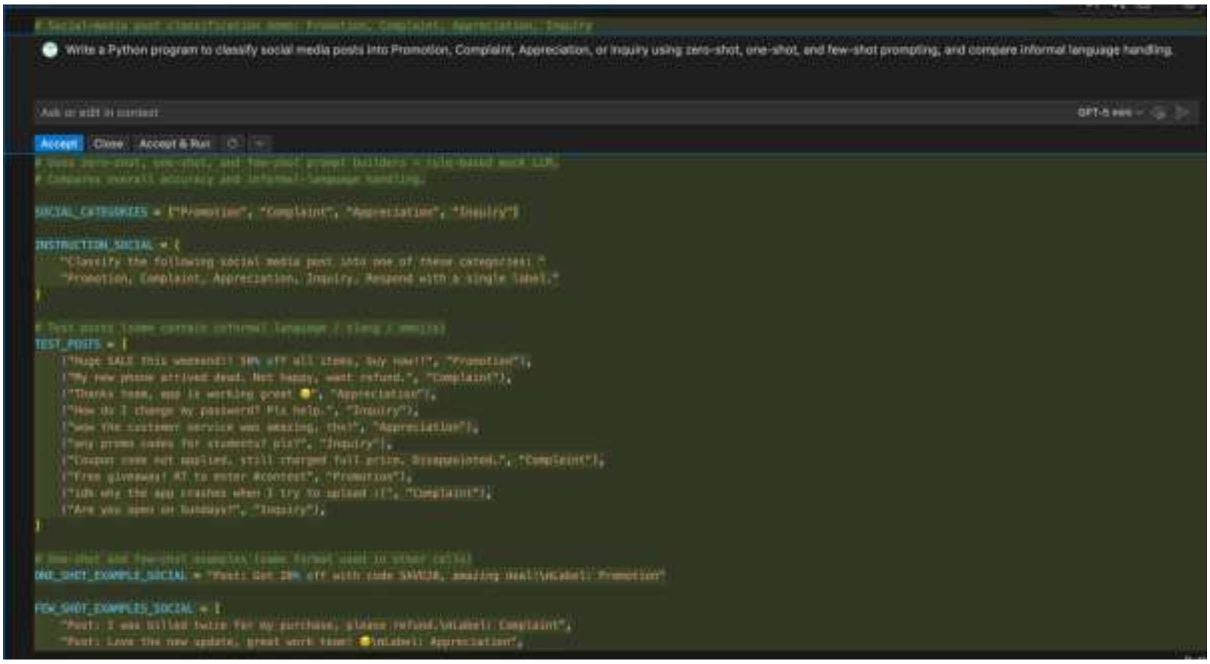


Output :



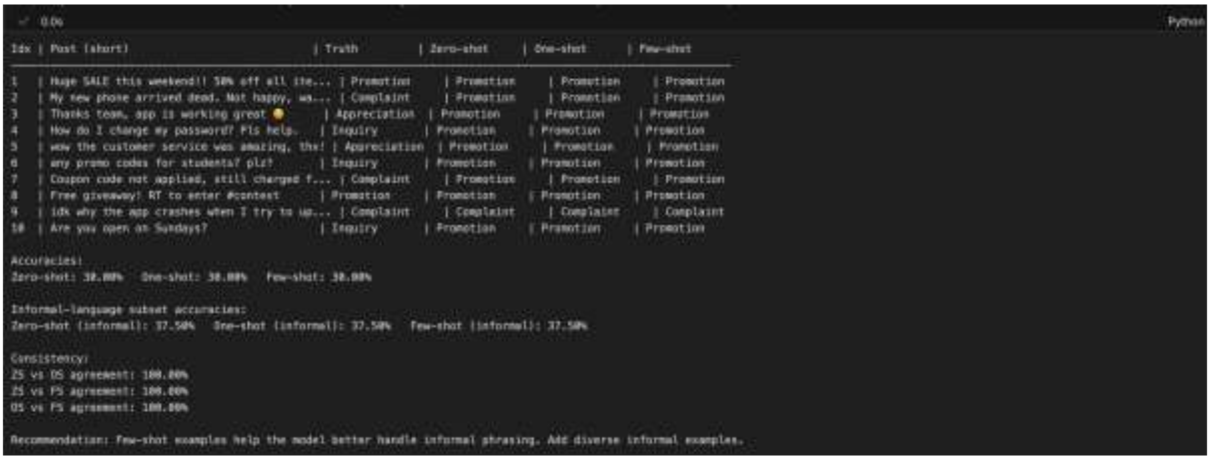
Task 4 :

Prompt : generate a python program to classify social media posts into Promotion, Complaint, Appreciation, or Inquiry using zero-shot, one-shot, and few-shot prompting, and compare informal language handling.



Social-media post classification demo: Promotion, Complaint, Appreciation, Inquiry
Write a Python program to classify social media posts into Promotion, Complaint, Appreciation, or Inquiry using zero-shot, one-shot, and few-shot prompting; and compare informal language handling.
Ask or edit in context
GPT-5 mini
Accept
Close
Accept & Run
# Uses zero-shot, one-shot, and few-shot prompt builders + rule-based mock LLM.
# Compares overall accuracy and informal-language handling.
SOCIAL_CATEGORIES = ["Promotion", "Complaint", "Appreciation", "Inquiry"]
INSTRUCTION_SOCIAL = (
"Classify the following social media post into one of these categories: "
"Promotion, Complaint, Appreciation, Inquiry. Respond with a single label."
)
# Test posts (some contain informal language / slang / emojis)
TEST_POSTS = [
("Huge SALE this weekend!! 50% off all items, buy now!!", "Promotion"),
("My new phone arrived dead. Not happy, want refund.", "Complaint"),
("Thanks team, app is working great 😊", "Appreciation"),
("How do I change my password? Pls help.", "Inquiry"),
("wow the customer service was amazing, thx!", "Appreciation"),
("any promo codes for students? plz?", "Inquiry"),
("Coupon code not applied, still charged full price. Disappointed.", "Complaint"),
("Free giveaway! RT to enter #contest", "Promotion"),
("idk why the app crashes when I try to upload :(", "Complaint"),
("Are you open on Sundays?", "Inquiry"),
]
# One-shot and few-shot examples (same format used in other cells)
ONE_SHOT_EXAMPLE_SOCIAL = "Post: Get 20% off with code SAVE20, amazing deal!\nLabel: Promotion"
FEW_SHOT_EXAMPLES_SOCIAL = [
"Post: I was billed twice for my purchase, please refund.\nLabel: Complaint",
"Post: Love the new update, great work team! 😊\nLabel: Appreciation",

Output :



0.0s
Python
Idx | Post (short) | Truth | Zero-shot | One-shot | Few-shot
1 | Huge SALE this weekend!! 50% off all ite... | Promotion | Promotion | Promotion | Promotion
2 | My new phone arrived dead. Not happy, wa... | Complaint | Promotion | Promotion | Promotion
3 | Thanks team, app is working great 😊 | Appreciation | Promotion | Promotion | Promotion
4 | How do I change my password? Pls help. | Inquiry | Promotion | Promotion | Promotion
5 | wow the customer service was amazing, thx! | Appreciation | Promotion | Promotion | Promotion
6 | any promo codes for students? plz? | Inquiry | Promotion | Promotion | Promotion
7 | Coupon code not applied, still charged f... | Complaint | Promotion | Promotion | Promotion
8 | Free giveaway! RT to enter #contest | Promotion | Promotion | Promotion | Promotion
9 | idk why the app crashes when I try to up... | Complaint | Complaint | Complaint | Complaint
10 | Are you open on Sundays? | Inquiry | Promotion | Promotion | Promotion
Accuracies:
Zero-shot: 30.00% One-shot: 30.00% Few-shot: 30.00%
Informal-language subset accuracies:
Zero-shot (informal): 37.50% One-shot (informal): 37.50% Few-shot (informal): 37.50%
Consistency:
ZS vs OS agreement: 100.00%
ZS vs FS agreement: 100.00%
OS vs FS agreement: 100.00%
Recommendation: Few-shot examples help the model better handle informal phrasing. Add diverse informal examples.